

FIG. 1

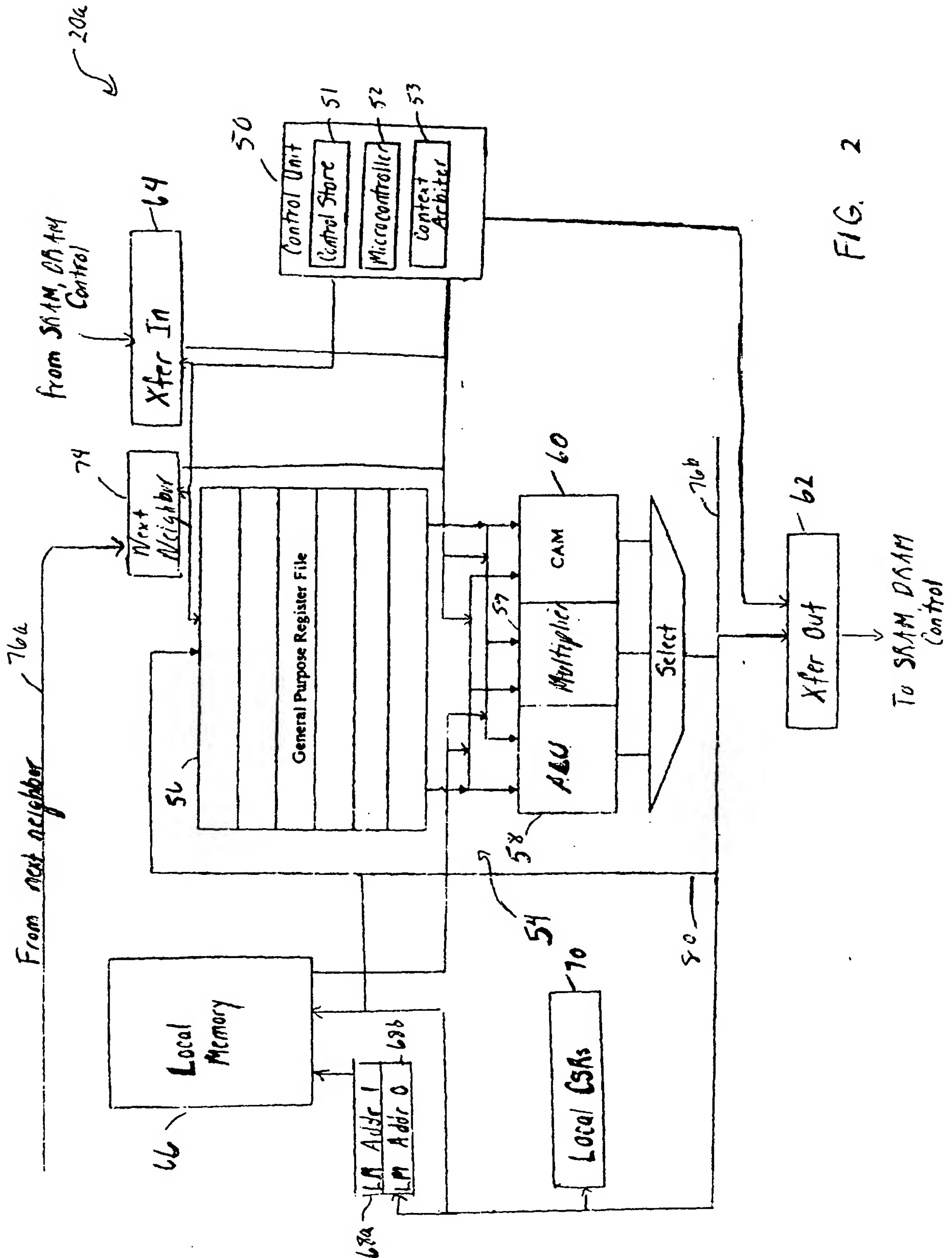


FIG. 2

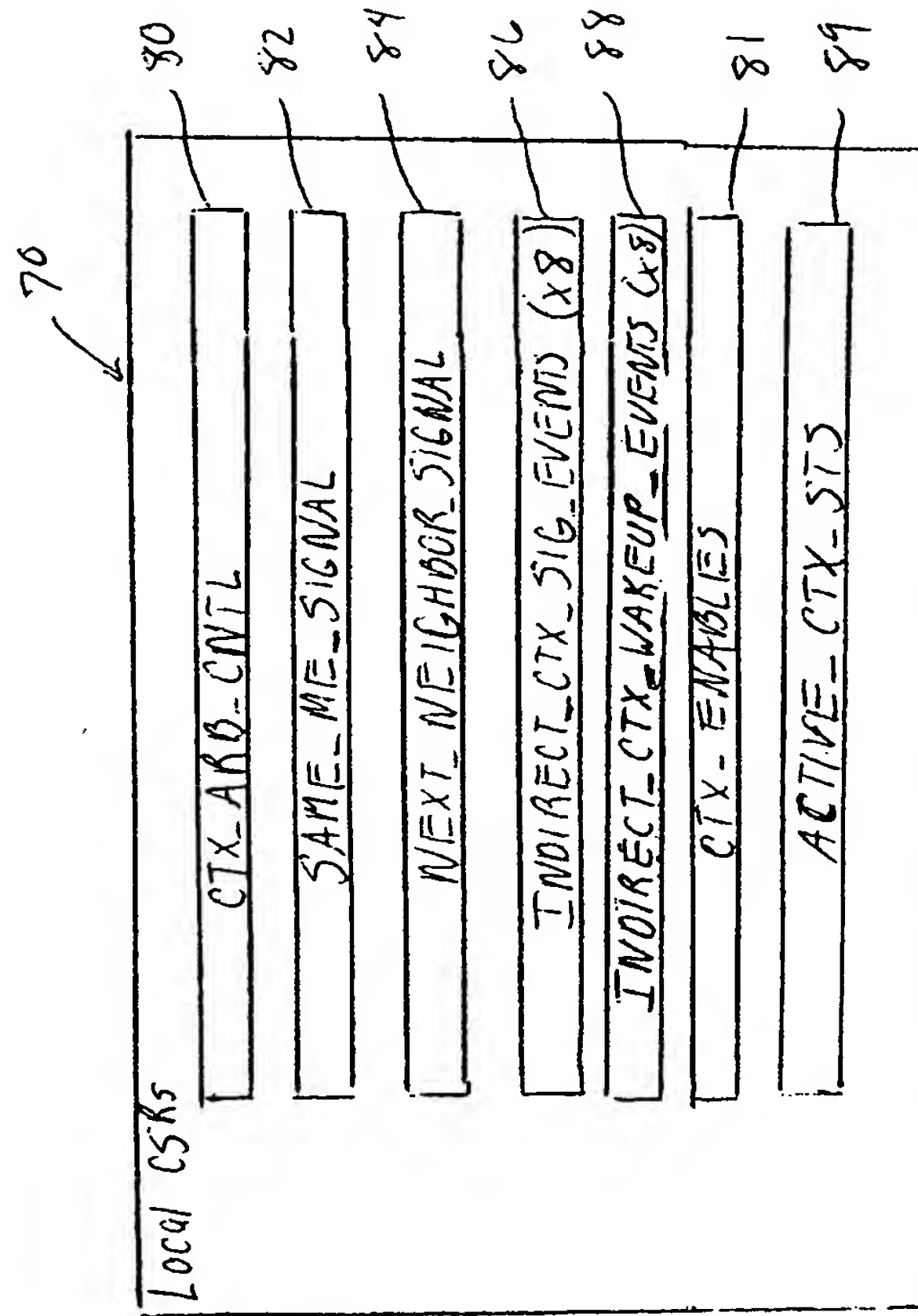


FIG. 3

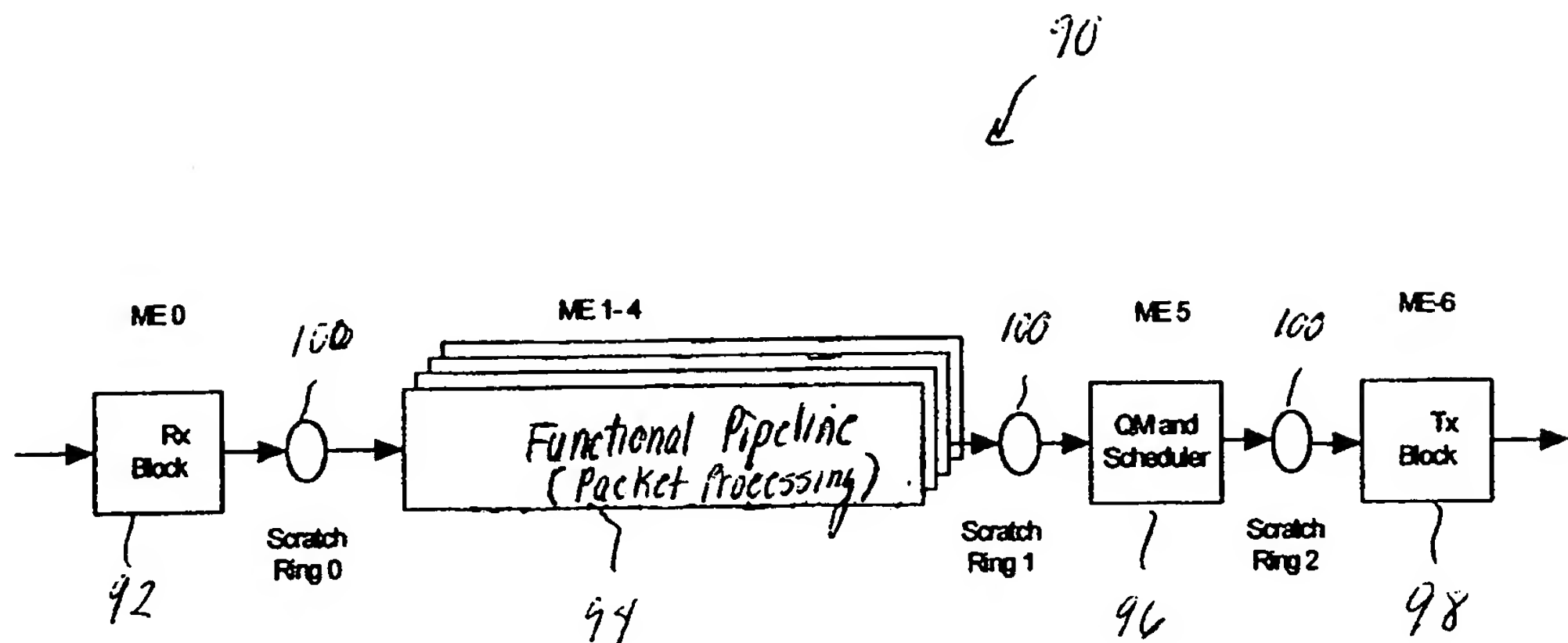


FIG. 4

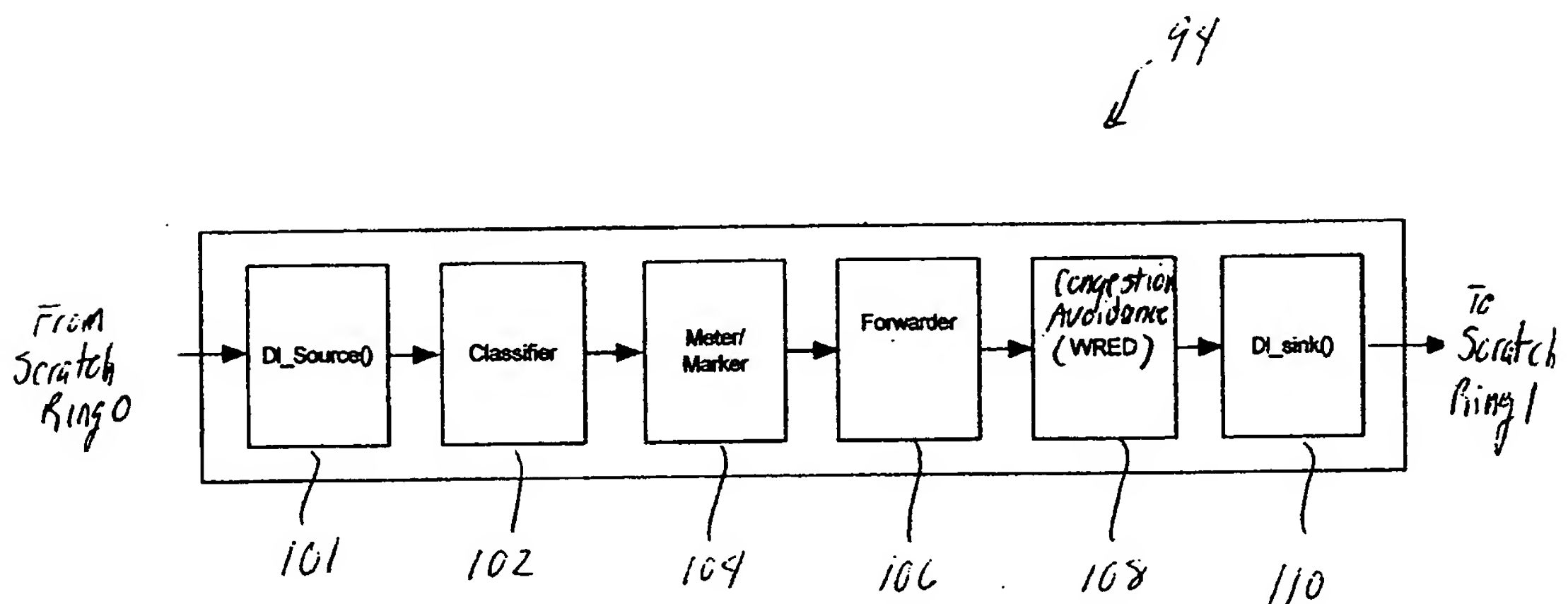


FIG. 5

126
↓

Register Name	Latency in terms of instructions			Usage Latency Comments
	Write	Read	Usage	
SAME_ME_SIGNAL	3	2	8	The same ME will be signaled 8 cycles after the CSR write.

122
/

FIG. 6

```
wred ()  
{  
    if (ctx () == 0)  
    {  
        // Wait for signal from previous ME and thread 7  
        wait_for_all (&next_thread_signal, &wred_next_me_sig);  
        cam_clear ();  
    }  
    else  
    {  
        wait_for_all (&next_thread_signal);  
    }  
    .....  
    .....  
    // WRED packet processing  
    .....  
    signal_next_thread () // Instruction 1  
    .....  
    // There is a minimum of 3 cycles delay between instruction 1 and instruction 2  
    // to allow the signal to propagate and to ensure thread execution sequence.  
    .....  
    // Wait for previous thread signal  
    wait_for_sig (&sig); // Instruction 2  
    .....  
    if (ctx () == 7)  
    {  
        // Signal next ME  
        cap_fast_write (wred_me_sig_csr, csr_interthread_sig);  
    }  
    else  
    {  
        // The thread gives up the context voluntarily at this point to ensure that  
        // thread 7 gets control as early as possible. If no context swap occurs  
        // here the thread would continue to execute non-critical section code or  
        // next microblock, thereby delaying thread 7 getting the control.  
        ctx_arb (voluntary);  
    }  
    .....  
    // Critical section processing ends  
    .....  
    // Non-critical section code or code for next microblock begins  
}
```

130

132

134

136

138

140

FIG. 7

150

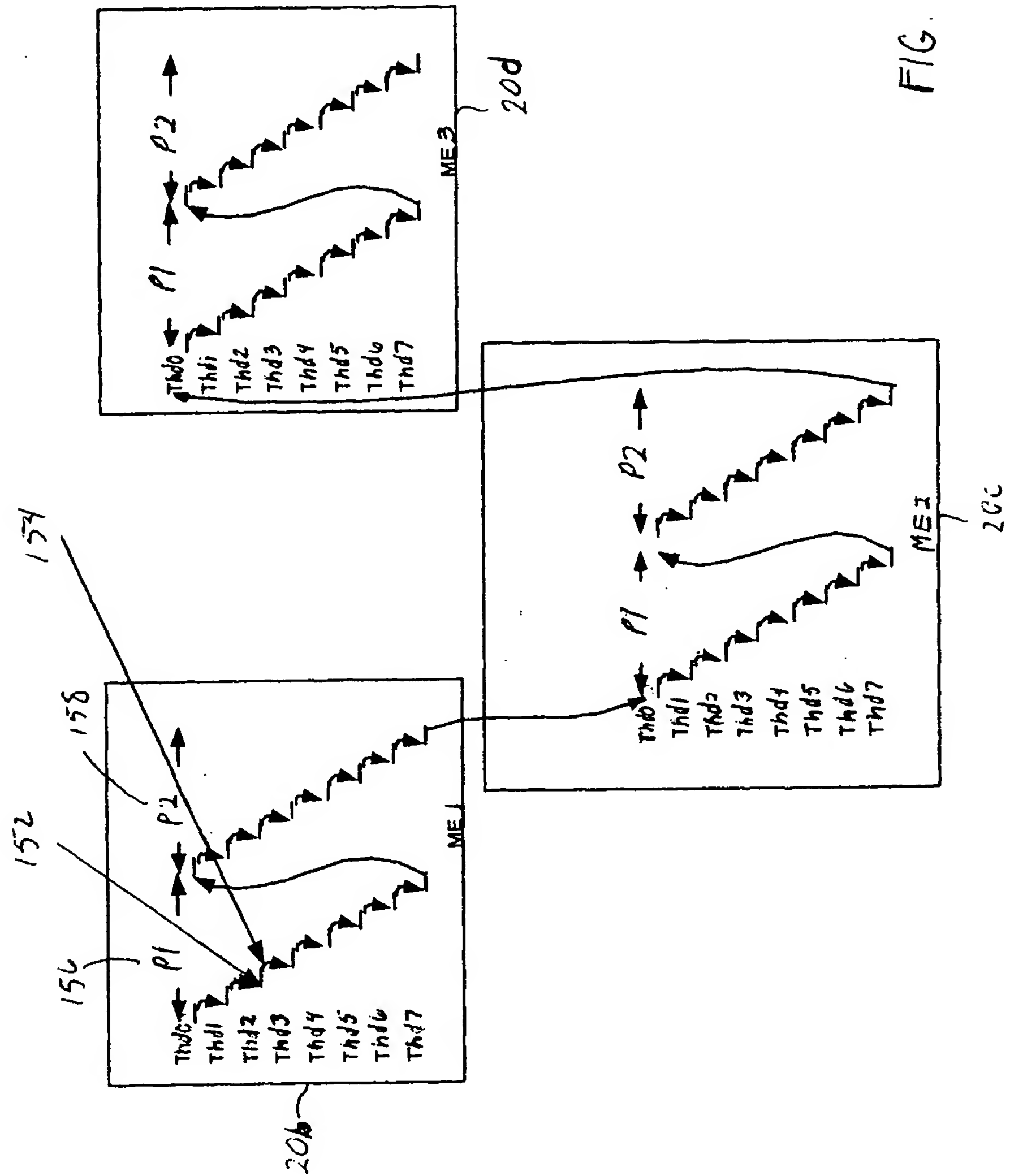


FIG. 8

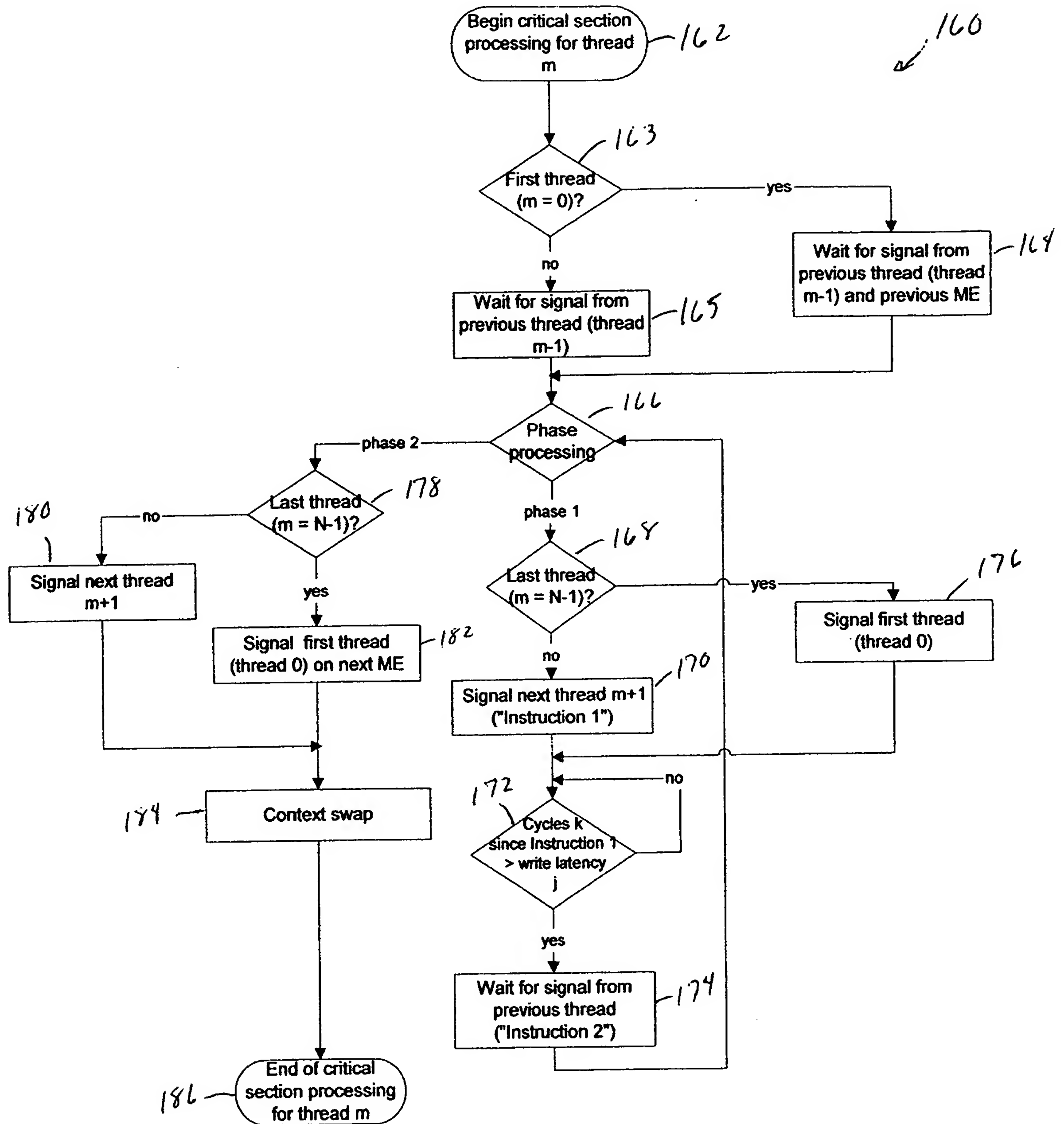


FIG. 9

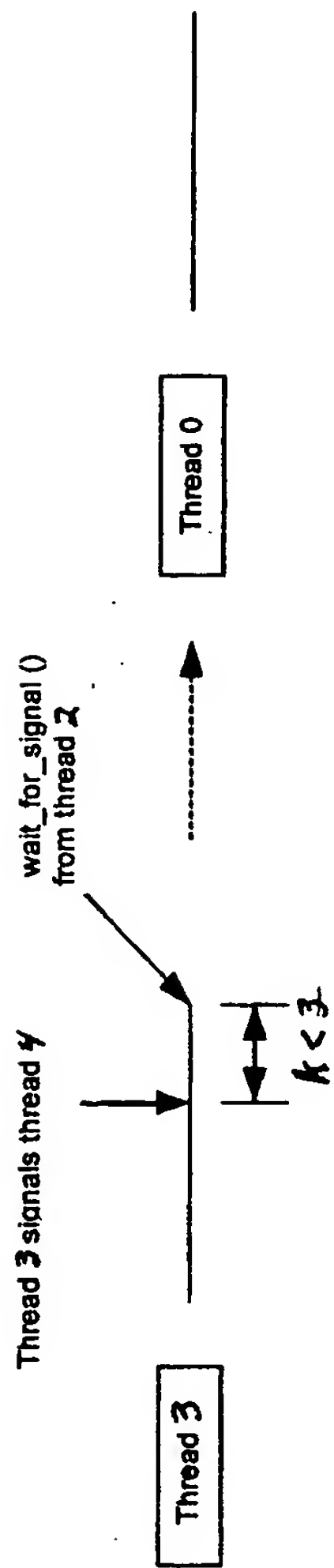


FIG. 10A

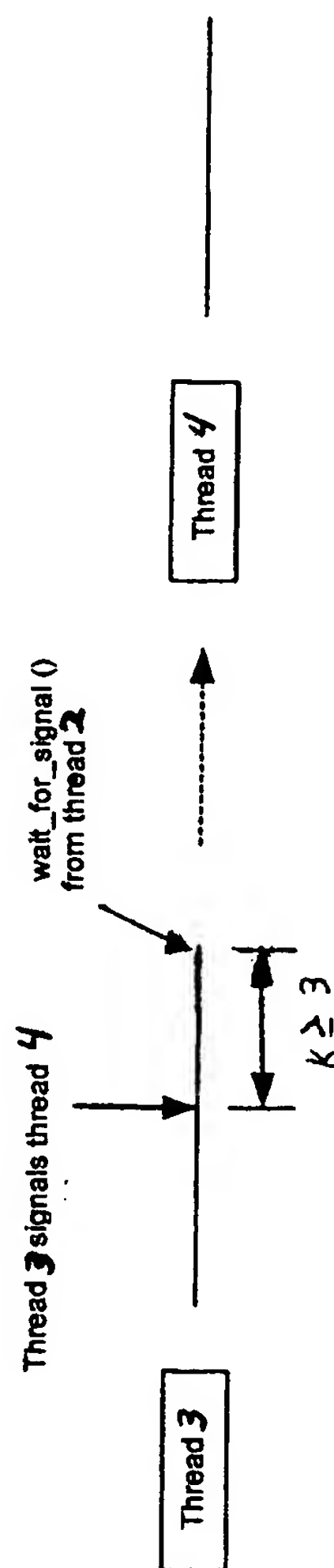


FIG. 10B

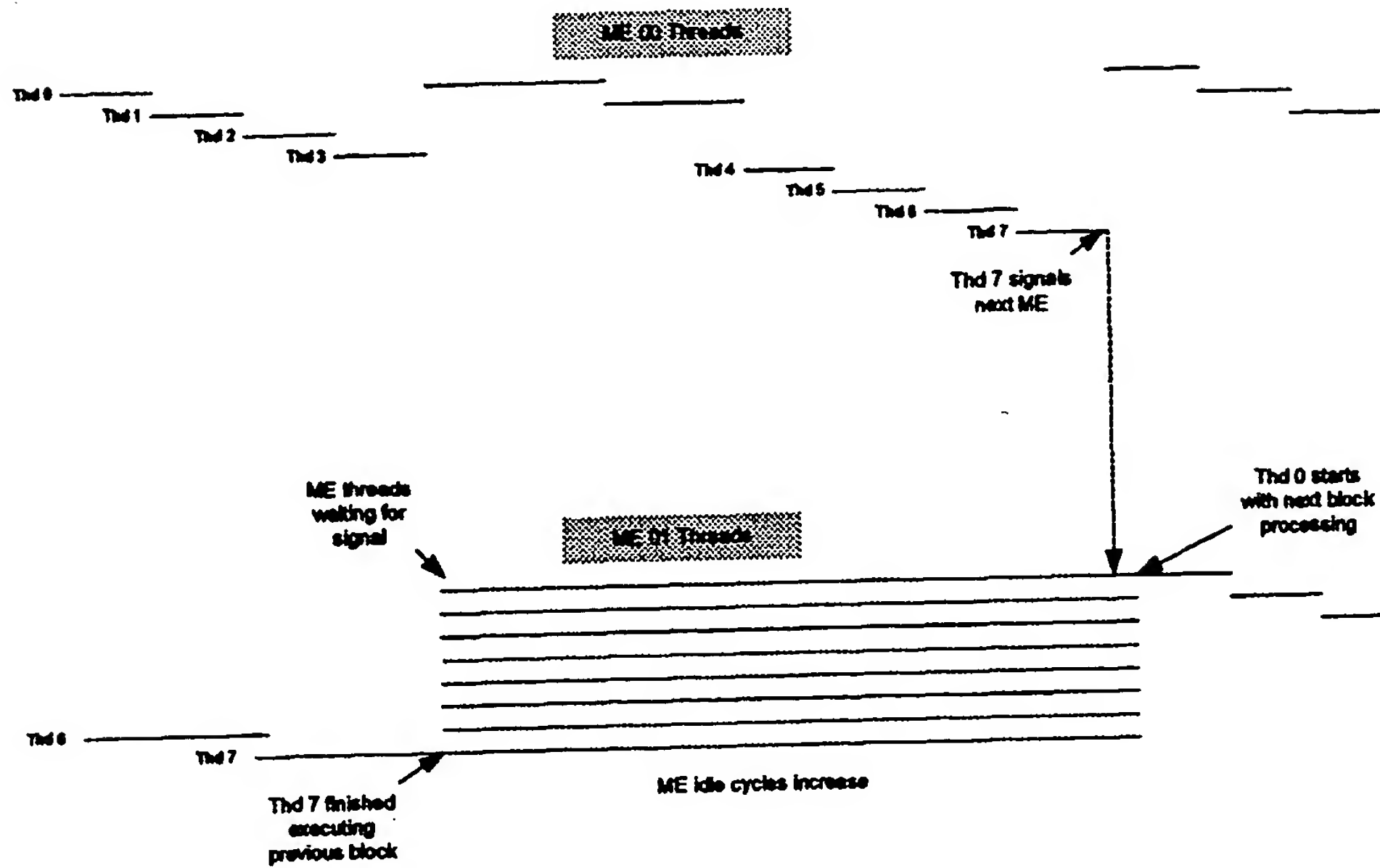


FIG. 11A

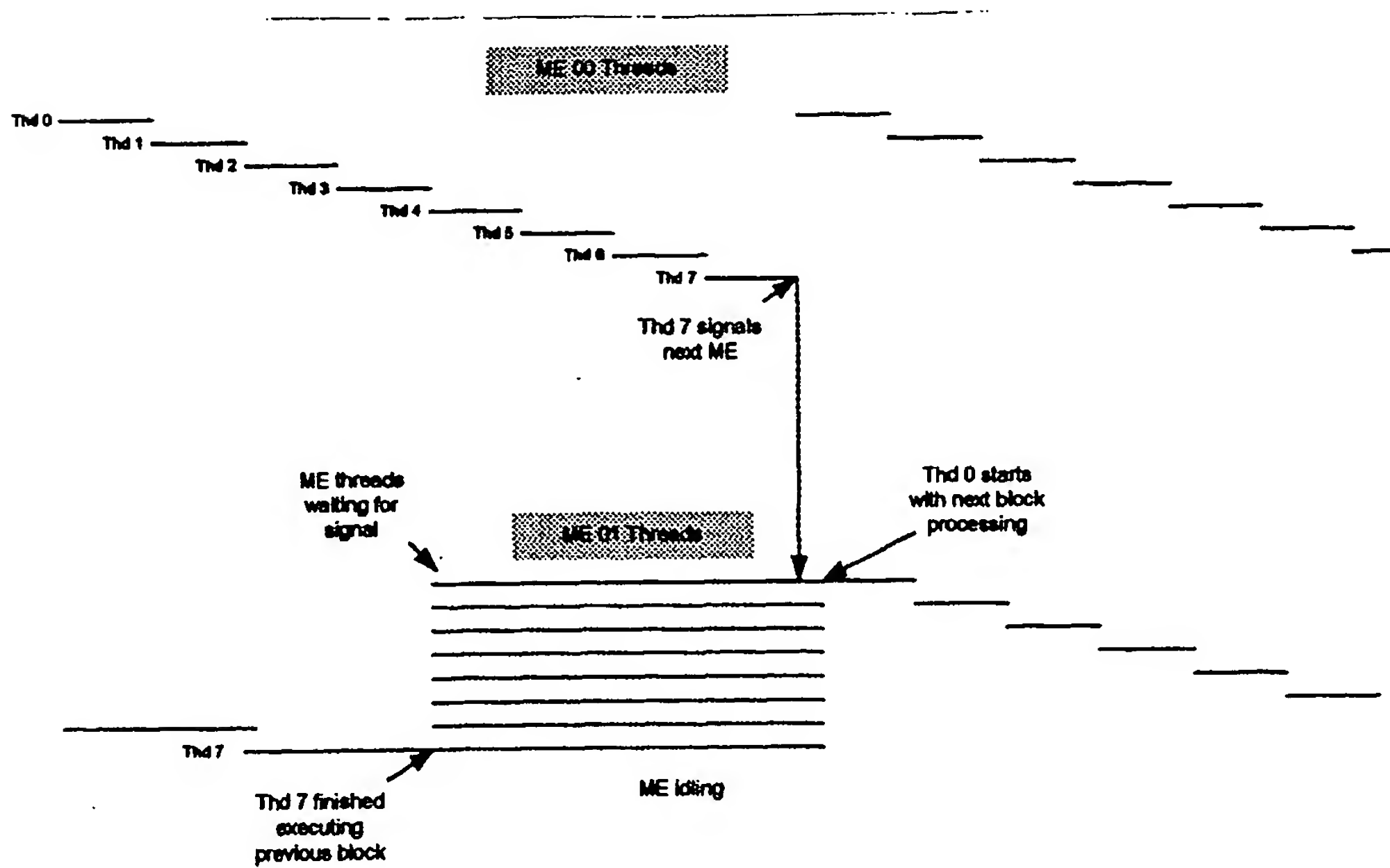


FIG. 11B

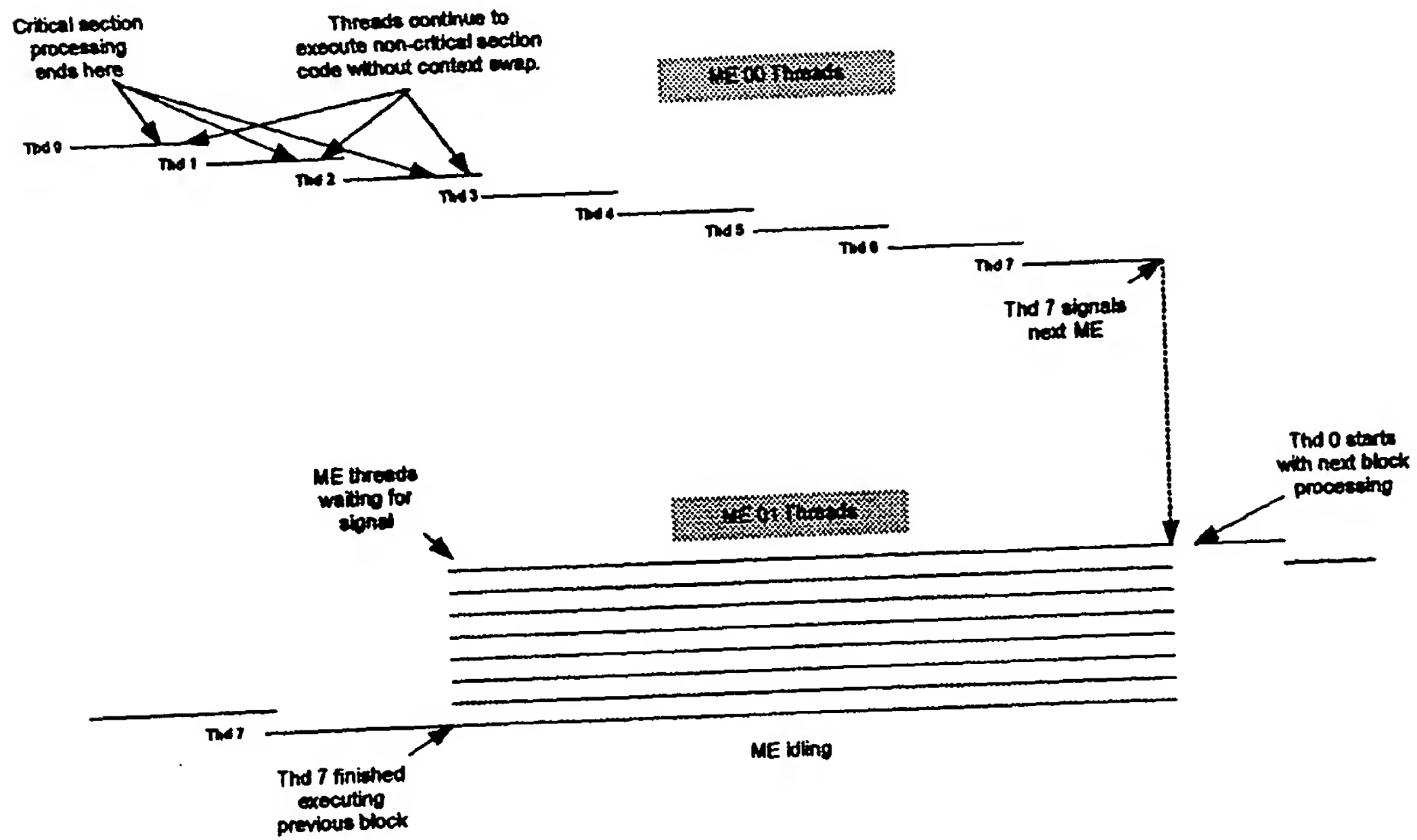


FIG. 12A

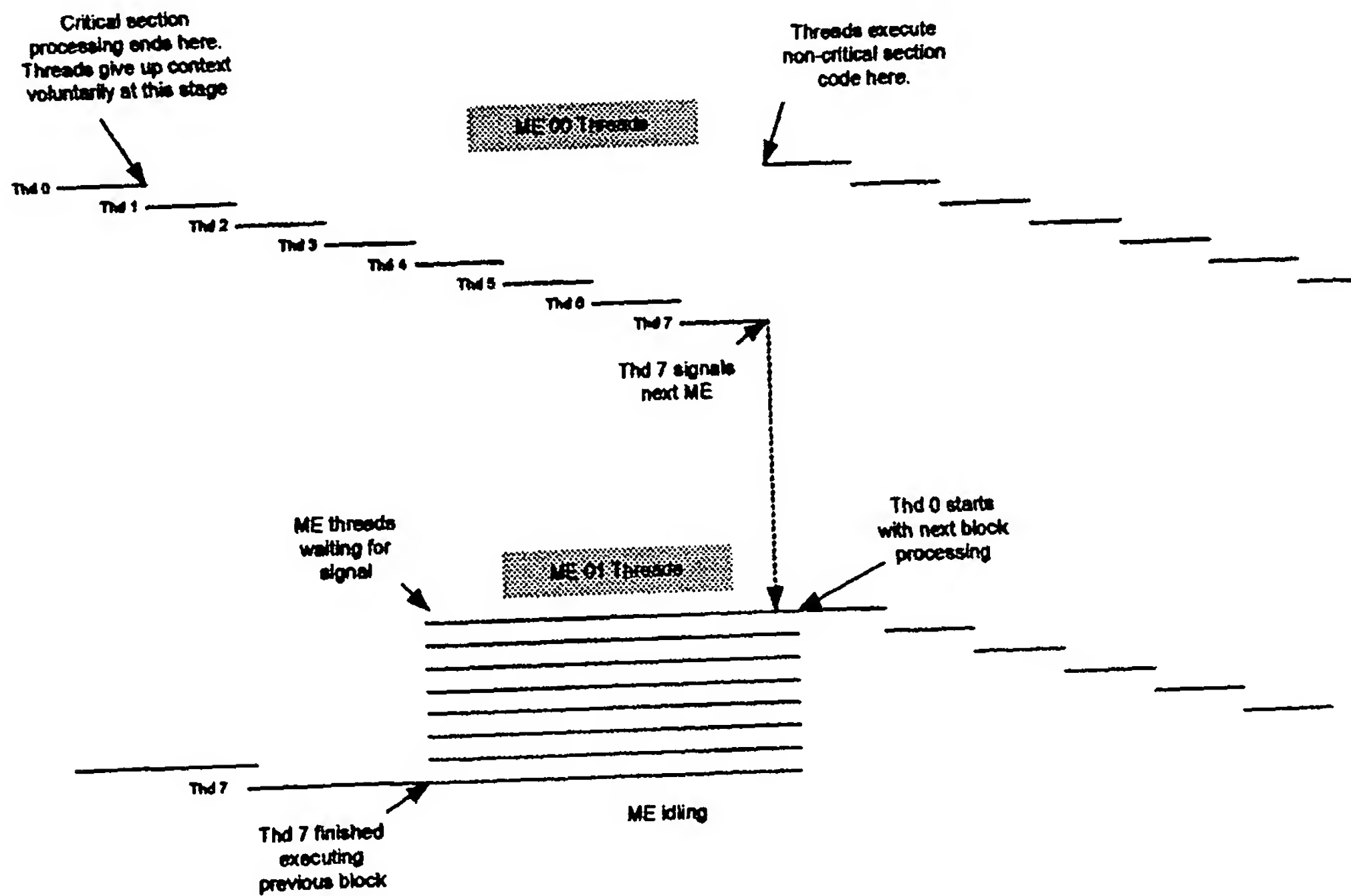


FIG. 12B

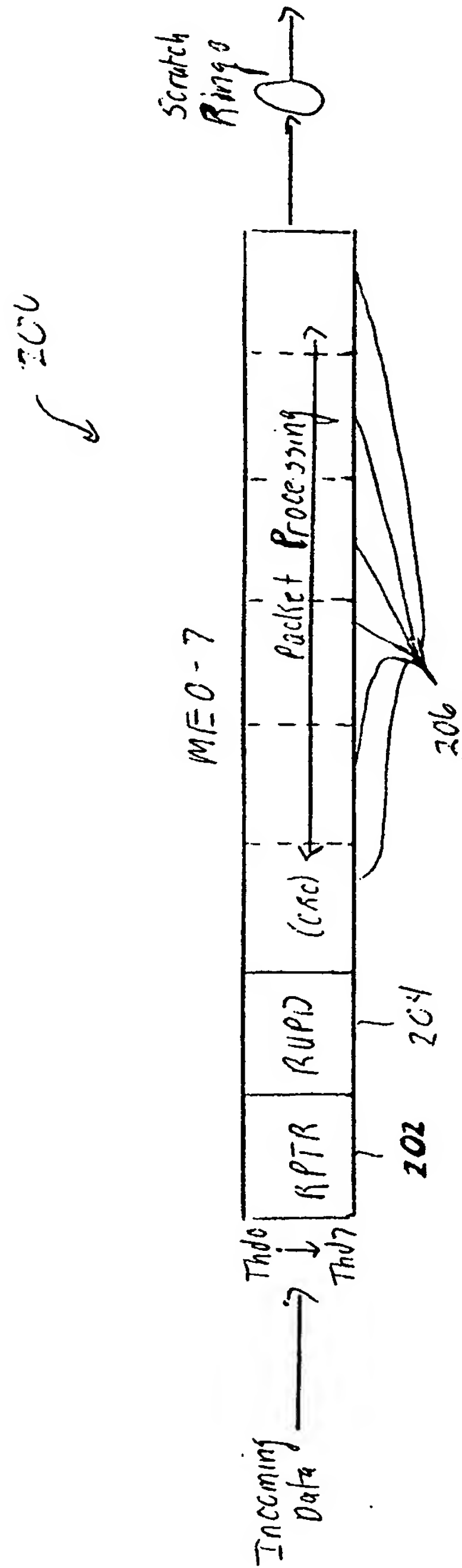


FIG. 13